

Fig 1

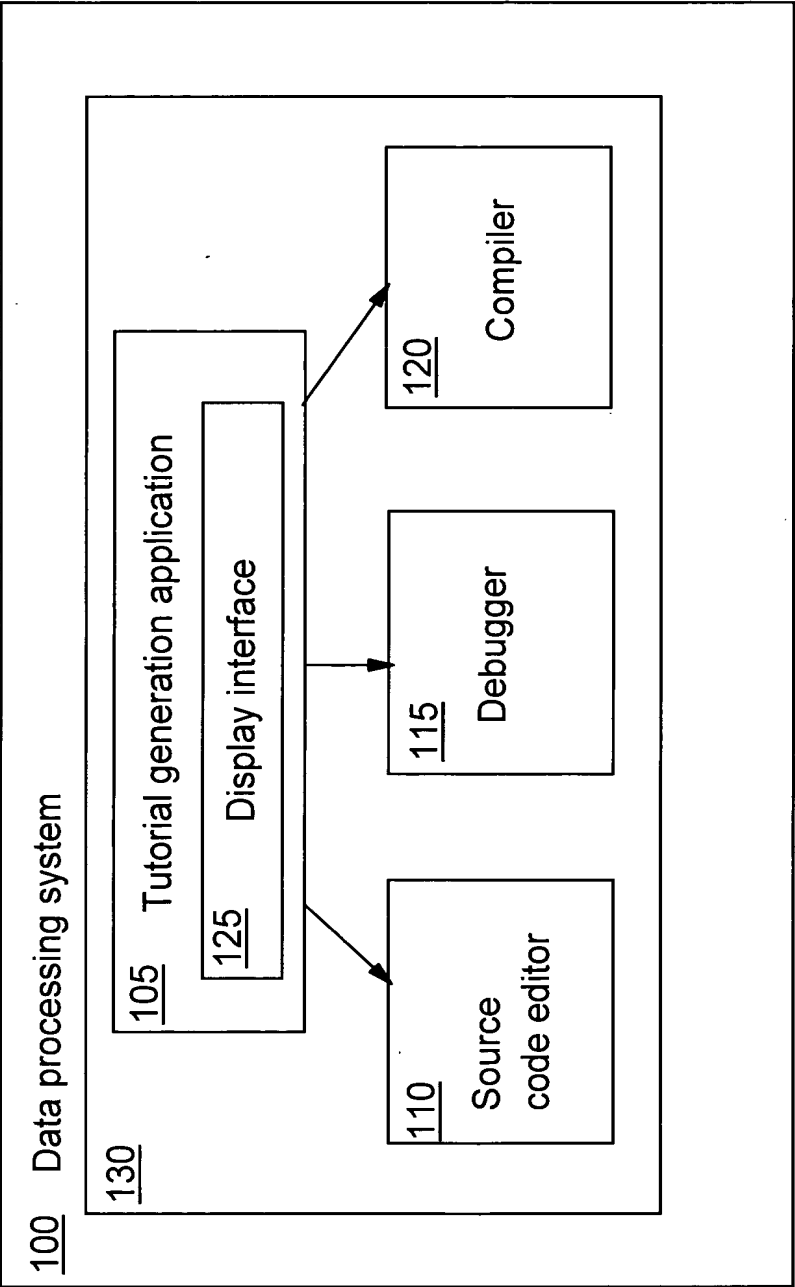
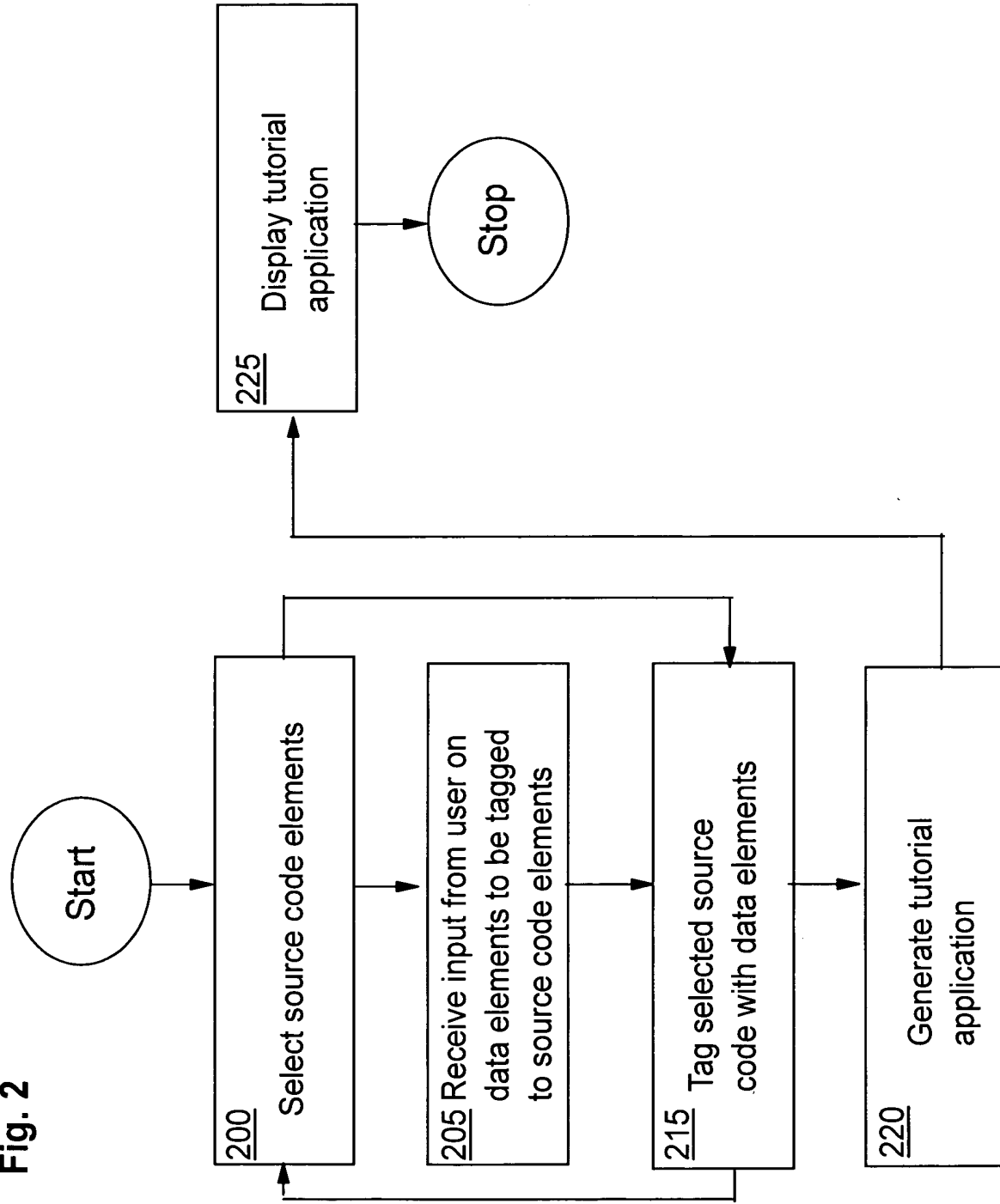


Fig. 2



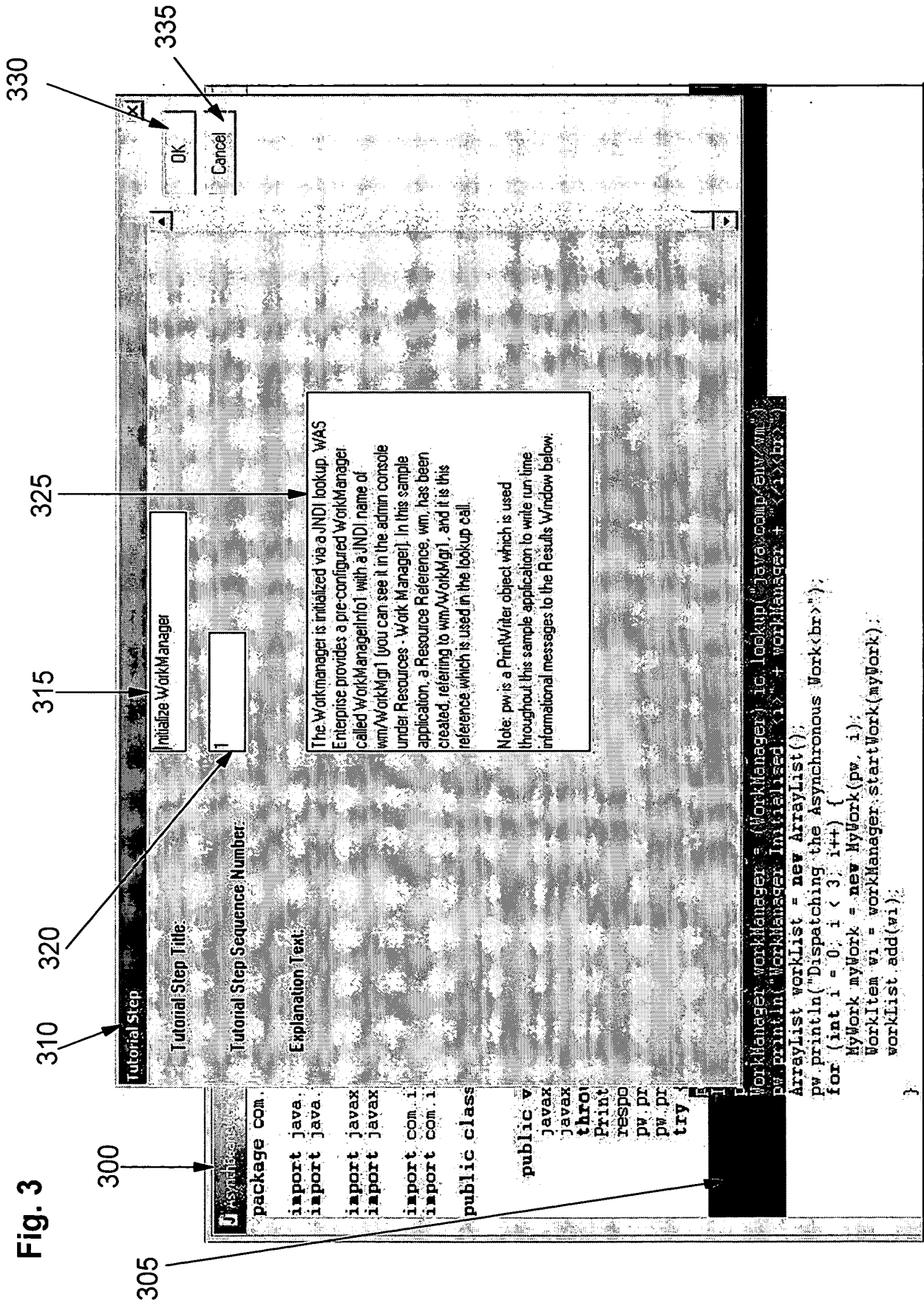


Fig. 3

Fig. 4

```

400      <TUTORIAL_STEP_SEQUENCE_NUMBER=1 TITLE="Initialize WorkManager">
405      <CODE_SEGMENT
410      pw.println("Obtaining InitialContext ...<br>");
415      InitialContext ic = new InitialContext();
      pw.println("Initialising WorkManager ...<br>");
      WorkManager workManager = (WorkManager) ic.lookup("java.comp/env/wm");
      pw.println("WorkManager Initialised: <i>" + workManager + "</i><br>");
420      <CODE_SEGMENT>
425      <EXPLANATION_TEXT>
      The WorkManager is initialized via a JNDI lookup. WAS Enterprise provides a pre-configured
      WorkManager called WorkManagerInfo1 with a JNDI name of wm/WorkMgr1 (you can see it in the admin
      console under Resources - Work Manager). In this sample application, a Resource Reference, wm, has
      been created, referring to wm/WorkMgr1, and it is this reference which is used in the lookup call.
      Note: pw is a PrintWriter object which is used throughout this sample application to write run time
      informational messages to the Results Window below.
430      </EXPLANATION_TEXT>
      </TUTORIAL_STEP>
435

```

Fig. 5

